PROFESSIONAL SERVICES

Creating Sustainable Business Advantage

PKI Architecture Document - Updated

July 6, 2000

Prepared for:
Ken Adrian
State of Iowa, IT Department
ITS 'B' Level, Hoover Building
Des Moines, IA 50319

Prepared by:
Professional Services
10 Fawcett Street
Cambridge, MA 02138

Table of Contents

1 Overview

Purpose of document

This Public Key Infrastructure (PKI) architecture document suggests a baseline, with major options, for the design of a certification-related system to support the use of public key certificates by the State of Iowa. Digital certificate functions will be provided by a Certification Authority (CA) and certificate authorization by Registration Authorities (RAs). This document provides data to help the State plan its PKI, set requirements for vendors, and estimate the cost of ownership of a CA. The CA can be operated by the State of Iowa using software purchased from a CA vendor, or the State can contract out the CA operation to a CA service vendor. This document is based on discussions with staff members from various State agencies, including the Information Technology (IT) department which is the customer for this consulting engagement. Requirements from these discussions and interviews are summarized in a PKI Requirements Assessment document that has already been delivered. The present document builds on this, and covers major options where it is not obvious which design option is best. A third document will analyze the capabilities of products from significant PKI vendors.

Overview

Starting from the requirements described in the first document, the simplest design that meets the requirements was determined and is presented as a baseline for analysis. This design has the lowest risk and least development, and should be adequate for the identified PKI system and applications requirements. The topics covered in the present document includes:

- The architecture of the CA
- The certification life cycle process
- Types of PKI-enabled applications that the State might use
- Methods for approving certificate requestors including identifying requestors
- Details of the required data fields and their processing
- Revocation and certificate revocation lists
- CA key management
- Miscellaneous issues

Major options are discussed in some detail, then summarized in table that summarize the issues, with pros and cons of the different options.

2 CA Description and Architecture

CA capabilities

Based on the earlier requirements analysis, the State of Iowa CA should include the features and options listed below if possible. Features of particular interest to the State of Iowa are described in more detail later in this document.

- Certification functions: Request input, authentication, generation, issuance, renewal, status checks, reports, revocation and CRL issuance
- Support for X.509 certificates including SSL (web), S/MIME (e-mail), and IPSec (VPN)
- Certificate authentication (approval): One manual method (Registration Authority) and one or more automated methods (e.g., pre-authorized data or automated connection to a back-end system)
- Certificate and user interface tailoring: Most web pages and data fields can be edited, and fields can be deleted, renamed, or modified to some extent.
- Support for all X.509 and most or all PKIX (IETF) certificate and CRL fields
- Hardware cryptographic units with secure backup/restore capability
- RA or CA originated revocation and reports
- LDAP interface to X.500 directories for storing certificates and CRLs
- Signed audit data
- Certification of other CAs, or certification by other CAs, via a PKCS #10/#7 interface

CA functions

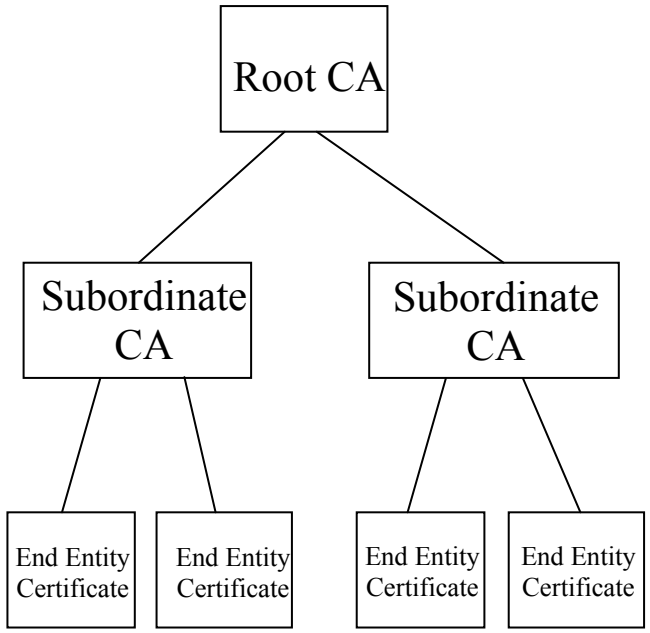
The CA handles the following tasks:

- Setting up a hierarchy of CA signing keys and certificates, with a State of Iowa root at the top (or possible tie-in to a commercial root whose certificate is already embedded in major applications).
- Processing certificate requests from users over a web Internet connection
- Obtaining approval from RAs run by State agency representatives or their designees
- Issuing approved certificates
- Maintaining the certificate database
- Handling status inquiries and reports
- Renewing certificates
- Revoking certificates on request and issuing a Certificate Revocation List (CRL)

CA hierarchy

Figure 2-1 shows a typical PKI hierarchy of CAs, with an offline (isolated) Root CA at the top, and subordinate to it the operational online CAs that actually provide certificates for users. The Root CA's primary function is to certify the CAs under it, and to pass the Root public key certificate to applications that must trust and rely on user certificates. The rationale for this segregation into multiple CA levels and related issues are presented later in a separate section on CA Key Management.

Figure 2-1 Sample Certificate Hierarchy
[view in Page Layout mode]



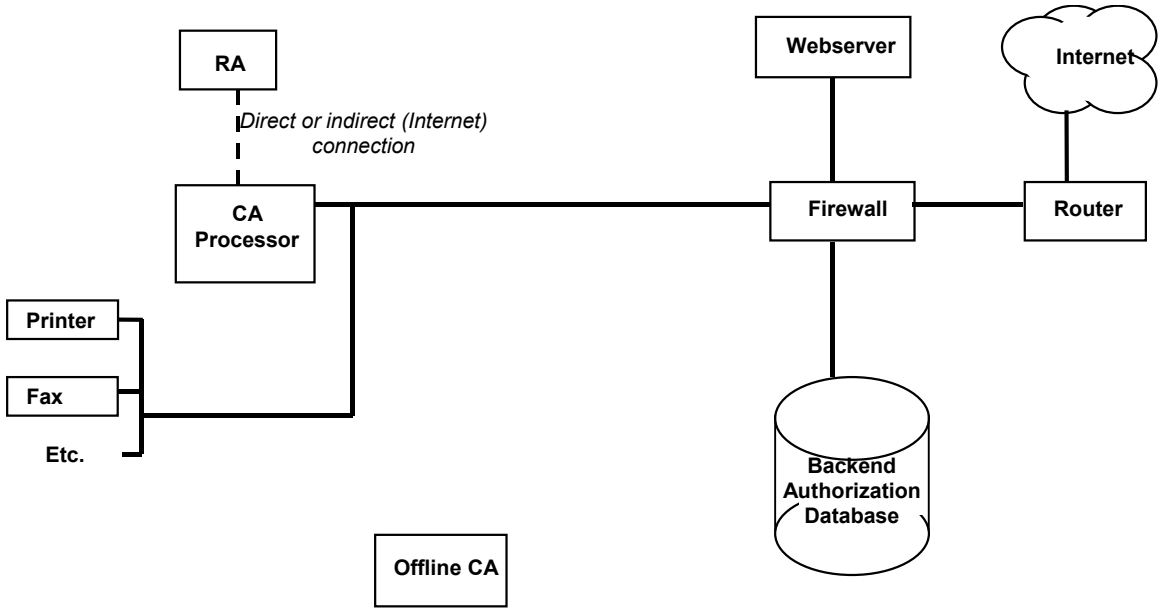
CA architecture

Figure 2-2 shows the primary components of a typical PKI system and the component interfaces. The components include:

- Public network, e.g. the Internet, for communications between the CA and user.
- Router: Standard router interfaced to the Internet.
- Firewall: Set up to only allow limited protocols entry into the CA web server from the Internet, as needed. The required protocols are HTTP (web), SMTP (email), and IPSec (VPN) traffic between the Internet and web server.
- Web server: For communicating with users and RAs over the Internet.
- CA processor: Performs most CA functions, including signing with associated cryptographic cards.
- Optional interface to external backend, if used for automated authorization.
- Other standard system components such as printers, disk packs, tape drives, etc.
- Registration Authority: A person or automated system, either co-located with the CA or communicating remotely over the Internet, that must authorize certificate requests before the CA can issue the certificate.

Figure 2-2 Typical CA System Configuration

[View in page layout mode]



CA interfaces

Users of State of Iowa PKI systems will connect to the CA using their standard web browsers or web servers, perhaps with plug-ins provided by applications vendors, or using similar clients. The CA's own web server interacts with users, providing them with certificate application forms to be filled out, and later with a link to download their certificates along with CRLs or CA certificates. Registration authorities use a CA interface, or if remote a web browser, to access special RA interfaces. All these web interfaces should be protected by SSL version 3.09 using X.509 certificates to support authentication and channel encryption. An email certification interface may also be used instead of a pure web interface.

The State may also be interested in a variant of this architecture, in which an RA or other State official obtains both a public/private key pair and a certificate for a user on the official's own PC (perhaps at a public kiosk), and approves the certificate request on the spot. The RA then provides the key pair and certificate to the user on a smart card or as a PKCS #12 file on a floppy disk. The end result is the same as for the first architecture above, but can be easier for some users. These issues are discussed at length in upcoming sections.

CA physical and operational architecture

Issues concerning the CA physical facility and its operation were already covered in section 9 of the Requirements Assessment document. Please refer there for details.

3 Certification Process

Overview

This section describes the complete process of obtaining a certificate, including identification, authorization and approval functions. It also discusses renewal of an existing certificate. The State is interested in supporting multiple methods of obtaining certificate authorization, matched to security requirements for the PKI-enabled systems being supported. Candidate CAs must therefore support multiple levels of identification and authentication.

Authentication options

Current CAs typically support multiple methods of authentication (obtaining approvals for certificates). These are summarized below.

- Registration Authority (RA): An authorized person at the CA, or at a remote browser with an SSL-protected connection to the CA, will download, review, and approve or reject certificate requests. The user typically logs on at a later time to download the certificate. This is a simple and very flexible method, but the involvement of a human RA limits its use to systems of a few thousand users.
- Pre-authorized: A file with authentication data is provided to the CA in advance by the user's organization. When the users later submits a certificate request, data in the request is compared against the pre-authorization file data. If the data matches, an approved certificate can be provided immediately to the user. This is a good system for intermediate sized systems, up to perhaps 10,000 users.
- Automated authorization: This is essentially an automated equivalent to an RA, where instead of a person, the authentication is performed by a program that interfaces between the CA and a customer backend system with authentication data. This normally requires some "glue" software, because every backend system is different, but once prepared this is the fastest alternative and is suitable for large systems.

Summary of certificate request authentication options

Option	Development	Advantages	Disadvantages
RA	None	• Available immediately • Very flexible	• Slow • Requires human action; limited to about 1,000 certificates
Pre-authorized file	Depends on CA chosen; may be minimal	• Can handle systems up to 10,000 or so certificates • Minimal to moderate development	• Somewhat inflexible
Automated "RA" program	Back end interface could take a few months depending on complexity	• Fast • Flexible	• Requires development of client (back end) interface software

Identification options

The State of Iowa also wishes to offer multiple levels of identification, where the RA or other authorizing entity can perform varying levels of effort to match the assurance level of the identification to the security level of the PKI-based system that certificates must protect. Three levels of identification discussed so far are:

- Face-to-face identification: This is likely to be the most secure case. An RA requires the user requesting a certificate to appear in person and present required documentation. The RA can also perform any other offline checks as appropriate. Some of the identification functions may alternatively be performed by another person, such as another State agency staff member or a notary public, who will pass the results on to the RA for final determination. Several Professional Services customers have used variants of this approach for high-security applications.
- Remote identification via data fields: In this case, the user does not appear in person as part of the certification process. Rather, he/she just logs onto a CA web page, and the CA web server triggers the user's browser to download a data form with fields that the requestor must fill out. Along with certificate data, the fields can include identification data that the requestor must fill out, which are hopefully known only to the correct user, such as social security number, date of birth, mother's maiden name, etc. When the RA reviews the application, this data must match agency records. This provides less security than a face-to-face meeting but is adequate for some systems, and is the simplest and quickest alternative, particularly for systems with a large number of certificates.
- Shared secret: This provides an intermediate level of security. Potential users and the CA/RA system share a password or other data, which is to be used when the user requests a certificate. This shared secret may be information provided by the user to the CA, such as personal data, or may be provided by the CA to the user, such as a random code. When the user later requests a certificate over the web, one of the fields that must be filled out is the shared secret. This tells the RA that the requestor is the correct user (assuming the user protected the shared secret properly). This alternative is often used in practice for large systems that must be automated but need more security than the pure remote identification above.

Shared secret issues

If a shared secret is used in the authentication process, it must be protected carefully. It should be communicated by a reasonably secure process, such as in person by the user's organization, via the U.S. mail, or by telephone, to minimize the risk of an intruder learning the secret. If RA-assisted certification is used as in the immediately preceding paragraphs, the RA might be instructed not to watch when the user enters the shared secret, although this is probably overkill since the CA must trust the RA in any case.

Summary of user (requestor) identification options

Option	Preparation	Advantages	Disadvantages
Face to face	Must designate State interviewer and set up procedures	• Available immediately • Very flexible	• Requires human action; may limited to smaller systems • Expensive to operate • Not clear who interviews user
Remote via data fields only	Choose data fields and train RA what to expect	• Fairly simple • Minimal to moderate development	• Least secure against spoofing (masquerading) attack
Shared secret	Determine and distribute shared secrets in advance of request	• Similar to remote option above, but more secure • Somewhat flexible	• Requires interaction with users to determine secret • Assumes potential users are known in advance

Overall certification process

A typical process for obtaining a browser certificate is described below. In this case, there are four processes, which are covered in detail in the following paragraphs.

- **Request**: The user requests a certificate from a CA customer web page.
- **Identification**: The RA or another person verifies the identity of the user.
- **Authorization**: The RA (or automated process) approves the certificate request. (This may be combined with the identification step above.)
- **Issuance**: The CA issues the certificate after the RA authorizes it.

Request process

- The user connects to a user web page at the CA when he wants a certificate. The State of Iowa may provide a hyperlink or button on any of its web pages, on which the user clicks to make this easier.
- The user clicks on a hyperlink to begin the certificate request process. The CA provides him with a form that he fills out. The form includes whatever certificate and authentication data is required. (Specific suggestions for this data are in covered in Section 5.) The user submits the form via a hyperlink or button click when finished filling out the form. The CA web server then instructs the user's browser to generate a key pair and to submit the public key along with the other signed certificate request data the user has filled out.
- The CA checks for correct format (syntax) of the request. It then verifies that the user's public key in the request body matches the signature on the certificate request, which is supposed to be created with the user's corresponding private key. The CA then holds the certificate until the information in the request can be authenticated.

Identification process

- For manual authentication, an RA is used, and face-to-face identification may also be used. This is the full case covered below. (An alternative automated process is optionally possible, as noted below in parentheses.) The CA notifies the user to wait or come back later, as appropriate.
- The user communicates with the State to provide or obtain information necessary for later certificate request authentication. For face-to-face identification, this is done by coming to a state official and providing necessary identification. This official may be the RA, or a person who communicates necessary information and/or approvals to the RA. The user may also be required to pay a fee at this time. (For the automated process, this contact may be via the Internet or e-mail. If no shared secret is needed, this step can be omitted altogether.)

Authentication process

- The authentication process established for this system is performed by an RA (or by an automated authentication system). These choices were summarized in an earlier paragraph. If a manual RA is used, the identification and authentication process can be performed together.

Issuance process

- When the certificate request is approved by the RA (or automated method), the CA prepares and signs the certificate.
- The user is notified when the certificate is ready. If a manual RA is used, this may be an hour or day after the request was submitted. (If an automated method is used, it can be just a few seconds while the user stays online.)
- The user downloads the certificate from the same web page as for the request, by clicking on a hyperlink or button to pick up the certificate.
- The user's browser is instructed to install and start using the certificate.

Face-to-face authorization timing alternatives

The preceding paragraphs assume the electronic certificate request occurs before face-to-face identification. Alternatively, the face-to-face component can occur first, before the user submits the electronic certificate request via the Internet. Now the overall certificate request process occurs in the following order:

- **Identification**: A State official or designee verifies the identity of the user.
- **Request**: The user requests a certificate from a CA customer web page.
- **Authorization**: The RA (or automated process) approves the certificate request. (This is thus separate from the identification step.)
- **Issuance**: The CA issues the certificate after the RA authorizes it.

The primary advantage of the original process (request before identification) is better security of the certification information. Since the user's private key is created early in the process, there is no chance for an attacker to subvert the process by masquerading as the user and substituting his own certificate information. Any attempt to do so will be detected later when the attacker fails to prove possession of the private key (by a signature, for example) that matches the certified public key.

An advantage of the alternative option (identification before request) is some protection against denial-of-service attacks against the CA database by flooding it with false certificate requests. Any false requests will be rejected quickly by the RA because prior identification has not been received. It also allows any payment to be

made at the beginning of the certification process. (Note: we know of no successful denial-of-service attacks against a CA to date, although it is possible that one will be attempted eventually.)

RA-assisted certification

Another variant of the certification process is when the RA helps the user obtain a certificate on the spot, at a State location, and gives the user the certificate in a secure manner to take home and install on the user's own computer. The process is the same as presented earlier, with two exceptions:

- The user's certificate request steps are performed at a State office or kiosk with an RA present, and the RA either advises the user what to do or does it for the user.
- The private key is created, and the certificate is issued (temporarily), to the RA's computer.
  - This can be done on the RA's computer (hard disk), and the private key and certificate exported onto a floppy diskette, which the user takes to his own computer and imports.
  - Alternatively (and more securely), the private key can created on, and the certificate issued to, a smart card or similar hardware token installed at the RA's computer, and the user takes this token to his own computer for installation. If a smart card is used, a card reader is needed at each location. A promising alternative is the use of small "dongles" which are plugged into the UCB port now standard on all new computers, without needing a card reader.

Then some additional final steps are needed:

- If a floppy is used, the RA exports the user's certificate from the RA's computer onto a floppy disk, using the standard PKCS#12 "export" function of current browsers. A password is required to protect the PKCS#12 file, and it is important that this password be entered by the user without the RA watching. Also, the private key and certificate must be deleted from the RA's computer using at least the browser's "delete certificate" capability. A hard disk active erasure program, which actually overwrites the hard disk surface with new data, may also be used for greater security. An alternative is to have the RA's computer generate the certificates on a removable crypto-device, which is removed after each use. The RA's computer should also be in a reasonably protected location.
- If a smart card or token is used, the RA's browser will just create the private key on the token, without ever writing it to the hard disk. This uses token vendor provided software, and perhaps a smart card reader, which both the RA and the user must have installed on their system. The token will require a PIN for any use, which the user should enter at the proper time without the RA watching. No deletion from the hard disk is needed. The RA then gives the user the token with the user's private key and certificate.
- The user then takes the floppy or token to his own PC, and loads it into his PC. If a floppy was used, the standard browser "install" function is used, with entry of the PKCS#12 file and password. If a token was used, the token is plugged into the reader and used per the token vendor's instructions, with entry of the PIN.
- The user's browser is instructed by its software to install the certificate,

which is then ready for use.

Because of concerns about possible loss of control of a private key, RA-assisted certification is probably best suited for a closed system where all parties agree to accept the risks (assuming the risks and procedures are discussed in contracts or a CPS). They are less suitable for a general-purpose signature-oriented system for citizens without an affiliation to a controlling organization.

Summary of additional authentication options

Option	Advantages	Disadvantages
Electronic request before face to face authentication	• More secure against "hijacking" request process (private key generated early) • Supported by most large CA systems	• Theoretical possibility of denial of service attack (flooding CA with false applications)
Face to face authentication before electronic request	• More secure against false applications • Can collect fee at start of process	• Some theoretical risk of attacker inserting his own electronic request in the middle
RA-assisted request and issuance via floppy or token	• Can be simpler for user • Lower level of problems with request process	• Less secure: RA knows more of user's request than normal • Not clear who will act as user-supporting RA, and where • RA's PC must be flushed for best security

Application usage of certificate

An application will later use the certificate, for authentication of the user or to obtain keys for signing or encryption. The CA plays no part in this process. The user generally has client software. This is often just a standard web browser, perhaps with plug-ins for additional features, or it may be a PKI-enabled secure e-mail package. The application with which the user is communicating is most often a web server with its own certificate, or it may be another user's secure e-mail application. Typically, an application server relies on the user certificate to identify the user, and the user browser or other client application relies on the application server certificate to authenticate the server.

- The user logs onto the application web page with his browser, which now has the certificate that it obtained from the CA above.
- The application server verifies the user's browser's certificate as part of standard RSA-based SSL processing. Alternatively, more advanced user client software offered by a third party vendor may be used.
- The application server should also check the latest CRL to see if certificate has been revoked. It checks by issuer (CA) and serial number.
- The application server may also use third party software to extract user ID data from the certificate to link to the backend database information about the user, and to make decisions about allowed processing (access control).
- From this point, the processing can be used to support whatever the system does, such as identity verification, access control, online purchasing, etc.

Renewal

Renewal of an existing certificate is supported by all major CA systems. It is much like the initial certificate request and issuance process, with some optional variations. It may proceed as follows:

- The user can be notified in advance that his certificate is about to expire. The time of notification and frequency of re-notification is generally a settable option.
- The user goes to a user web page, perhaps where he got the original certificate.
- The user clicks on a hyperlink or button to begin the renewal process.
- The user is provided with original data to review. He may be forbidden to change data..
- The user clicks on a hyperlink or button to submit the renewed certificate request.
- The CA commands the browser to provide a new key. The old key may be used only to prove continuity with the old certificate, for example in setting up the SSL session which delivers the new certificate request. (CAs may alternatively support using the old key for everything, which may be more convenient but is less secure. The State indicated it is not interested in considering this alternative, so it is not discussed further here.) Both the new public key and any use of the old key are verified by the CA.
- The user-to-CA SSL link uses the old key, which is therefore verified by the web server as part of the SSL protocol. The CA may use this to verify that the user has the old private key and tie this request firmly to the original certificate
- The request is signed with the new key. The CA verifies the signature against the new public key in the certificate request.
- The CA may be set up to wait for authentication, or may just provide a new certificate based on the key authentication above (no new authentication).
- The user downloads and installs the new certificate from the link on web page, just as he did with the original certificate.

The certificate lifetime (from “not before” to “not after” date/times) can be specified for each system when its CA is set up. Typical certificate lifetimes are from one to four years. Maximum certificate lifetimes for State certificate classes will be identified in the certification practice statement.

4 CA and User/Organization Functions

Overview

This section provides a high-level summary of functions of the primary participants in the certification process: the CA, the user organization (such as State agencies), and users. Their duties to systems that use the certificates are not covered in detail. This document only provides a high-level overview. Details are normally provided in a certification practice statement (CPS), which would expand on some of these topics and should be considered the official statement for such topics.

CA operator responsibilities

The State IT Department (ITD) or a CA service vendor will act as the CA for the State systems, using CA vendor software product and other supporting hardware and software. The CA duties may include:

- Operate the CA service out of a secure facility.
- Receive identification information provided by or to the user.
- Receive certificate requests from the user.
- Provide the certificate to the user after approval by the RA.
- Maintain a database with certificates and audit data.
- Provide reports and status summaries to customers as needed.
- Provide help desk support to users who are having difficulties. (CA product vendor staff members can help with exceptional difficulties.)
- Revoke a certificate when directed to do so by an RA or other authorized officer. Include the reason in the database.
- At all times, protect any sensitive user data that the CA must hold during the certification process.

User organization responsibilities

The organization that represents the user (such as a State agency) and runs the PKI-enabled applications should perform the following functions.

- Notify users where to obtain a certificate.
- Provide data needed by the users and/or the CA. This may include:
  - Any shared secret data that the CA must know, for example for a pre-authorization data file.
  - Other data that the user would not know before certification.
- Authenticate and authorize certificates for its users using one or more of the authentication methods, for example by acting as an RA (or outsourcing this responsibility).
- Link to customer databases or back end systems as needed to verify certification data.
- Review user-provided data for completeness, accuracy, and authentication checking.
- Add any additional data.
- Request revocation of certificates.

User responsibilities

The user's responsibilities relating to certification include:

- Obtain a certificate
- Protect the shared secret, if used
- Use the certificate only for approved purposes
- Protect the private key
- Report any events that could require the certificate to be revoked.

Protection of the private key is critical, because otherwise the certificate cannot be trusted. The user generally must be required, by law or contract, to protect the private key if he wishes to participate in a PKI-enabled system that replaces traditional paper and handwritten signature processes.

5 Certificate / Request Data Fields and Processing

Overview

This section suggests how to provide the basic certificate and certificate requests fields defined in the PKI Requirements document. Certificate request fields use some available HTML (web form) fields of the CA-to-user interface. Certificate fields use X.509 fields, mostly in the subject (user) Distinguished Name (DN). There may also be some extension fields, for example applications may use certain extensions. (The Subject and Issuer DNs are fairly simple fields near the front of the certificate, where they are easy to access and extract, and are not considered extensions.) The fields are somewhat uncertain at this point and may vary with different applications, but they can easily be changed if necessary to reflect changes in requirements. Special tailoring methods are generally available from CA product vendors to select which fields, text and images are used

Certificate request fields

Possible certificate request data is discussed in the PKI Requirements document. It is provided to the CA by the user via a web form with many available fields. A given CA will only use the fields it needs, and may not even choose to present them on the user request page. Any fields left blank by the user should also be left out of the certificate by the CA software, unless they have been tagged as required during the subordinate CA setup, in which case the request will be rejected if they are left blank. Typical certificate request fields that are available are shown below. A field can be for data that is expected to go into the certificate, in which case it tracks the X.509 data fields rather closely. Additional data that the CA requires but which is not necessarily included in the certificate is noted as "(auth data)" below.

User data	Possible internal fields to use
Name	First, middle, and last names
Email address (if S/MIME cert)	Email address
IP address (for some VPN certs)	IP address
Social Security Number	Social Security Number (auth data)
Birth date (option)	Date of birth (auth data, maybe in cert)
Employee number (for example)	Organizational ID (auth data) [1]
User organization name	Subject DN Organization [1]
Other user organization data	Subject DN Organization Unit(s) [1]
Shared secret	PIN (auth data)

Notes

1. The organization, organizational unit and organizational ID number information are intended only for *organizational* certificates, which may used as signing certificates for agency or company officials, for example. The organization information may be placed in one or the Organizational Unit fields. In *individual* certificates for personal use, these fields should either be left blank, or filled with a default value such as "Personal certificates only". The RA must ensure that certificates have the proper value in these fields.

Certificate fields

The basic certificate data is listed in the PKI Requirements document. It is provided by the user in the certificate request form, by the RA or other authentication mechanism, or perhaps by the certificate request web page itself. The data is placed in one of the fields defined by the X.509 certificate standard, such as one of the Subject (user) Distinguished Name ("Subject DN") fields. A given CA will only use the fields it needs. Fields that could be used are shown below.

Data	X.509 field(s)
User name	Subject DN Common Name
User organization name [1]	Subject DN Organization
Other organization data, if needed [1]	Subject DN Organization Unit(s)
Birth Date (in case of DN collision) [2]	Subject DN Date of Birth (DOB)
Public ID number (option?) [3]	Subject DN Organization Unit(s)
Country = "US" [4]	Subject DN Country [4]
State = "IA" [4]	Subject DN State or Province [4]
Issuer [4]	Issuer DN Common Name [4]
(e.g., State of Iowa Certification Authority)	
Possible extensions, as needed: [5]	
Email address (if S/MIME cert)	Email address in Subject Alternative Name
URL (web location) for CRL	CRL Distribution Point
URL for policy or CPS	Policy: CPSUri
Other extensions [6]	*[As appropriate]*

Notes

0. See notes in "Certificate request fields" section above for notes on fields derived from the certificate request.
1. The organization data is needed only for certificates intended to work with a particular organization's (e.g., agency's) systems. If the State decides to offer personal certificates not related to any specific organization, a generic organization such as "State of Iowa Personal Certificate" can be provided by the CA or RA, and no Organizational Unit fields should generally be used.
2. The birth date allows the CA to distinguish among duplicate user names.
3. The Social Security Number must be protected, and thus cannot be placed in the certificate. A substitute number (such as Student ID) may be available in some cases, and if so it can be put into a spare Organizational Unit field.
4. The country and state codes and Issuer information should be filled in by the CA or RA rather than the user.
5. Extensions may be needed for proper operation or to meet standards.
6. If additional data is needed for which appropriate standard extensions are available, it can be placed in additional Subject DN Organizational Unit fields.

Extensions	The State of Iowa is expected to minimize the use of unnecessary certificate DN fields and extensions. However, some fields or extensions may be necessary, for example for browsers to work well, to meet organization's requirements, or for additional information. Extensions are discussed in the "Revocation and CRLs" and "Multi-Certificate System Design" sections of this document.
Server certificates	SSL web server certificate are much like browser SSL certificates, with the addition of some special extensions used for servers. The exact format differs by server type, although a basic format can be used to support both Netscape and Microsoft servers if necessary. We recommend that a standard CA-provided server certificate template be used, since they have been refined by CA vendors to work well with Netscape or Microsoft servers.
Mandatory fields	Certain fields can be marked as mandatory in the certificate request file. This means that they cannot be left blank. This should be used carefully if at all, because it means that every certificate for that CA must use those fields. This should be discussed with certificate format experts before marking any fields as mandatory.

6 Revocation and CRLs

Overview

This section summarizes matters relating to revoking an invalid certificate and issuing a current certificate revocation list (CRL). A user's certificate may become invalid because information in the certificate changes, for example a user leaves an organization. More seriously, a certificate can become bad if the user or the certifying organization think that the private key may have been exposed.

Revocation process

Certificates may be revoked by a CA operator or RA, in response to a request from the user or an authorized customer officer such as an RA. The operator may revoke when a user indicates that the certificate is no longer valid, or may decide based on other information. The result is that the certificate status entry in the CA database is marked as "revoked". This interface should allow a revocation description to be entered into the database, which can be pre-selected choices or any text the CA operator wants to add.

Before revoking a certificate, the CA operator or RA should demand some form of authentication from whoever requested the revocation. The exact process and authentication data, and who is authorized to cause a revocation, can be determined by the State agencies and user organizations, and documented in the Certification Practice Statement. As an example, the CA may require one or more of the following: a fax of a signed request on company letterhead stationery, use of a pre-set password, a telephone call, etc.

CRL issuance

A certificate revocation list is not automatically issued when a certificate is revoked. The CRL can be generated manually by CA operator, or can also be generated automatically on pre-determined periodic basis. The CRL is issued by the same CA that originally issued the now-revoked certificate(s).

CRL posting and download

Once a CRL is generated, it will be posted to a web page for interested parties to download. Relying parties must verify the CA's signature on the CRL. Only the most recent CRL from that CA should be used. The URL of the web page can be provided to interested applications that might use the certificate, which should download the latest CRL periodically and check all certificates when presented to see if they have been revoked. The CRL download link can also be inserted in either RA or user home pages when the CA is set up.

LDAP interface option

A subordinate CA can also be set up to provide CRLs to an X.500 directory via a standard LDAP interface, similar to the way certificates can be presented to a directory.

Optional revocation checking and OCSP

As an alternative to CRLs, certificate validity can be checked by using a trusted server. Most CAs have partnered with ValiCert, the company that dominates this market, to provide such support, and have successfully tested that the CA's CRLs are compatible with ValiCert's service and products. Most CAs (and ValiCert) now support the OCSP (Online Certificate Status Protocol) interface for certificate validity checking. The CA or ValiCert can act as OCSP "responders" (servers) to respond to an OCSP client's request for status information about a specific certificate. The CRL is the most secure mechanism for the CA to provide revocation information to an external agency such as ValiCert, which can then process the information appropriately.

Temporary certificate hold

Most CAs also allow a certificate to be put on hold (suspended) temporarily, until it can be determined that whether the certificate should actually be revoked, for example while waiting for authentication of a revocation request. If a CRL is issued while the certificate is still on hold, it will be included in the CRL. If the hold is removed, however, the certificate status returns to "active" and it is not included in future CRLs. This certificate hold capability may be appropriate to use if the State is notified of a possible revocation situation other than by a user, and does not want to permanently revoke it until further investigation.

Summary of revocation-related options

Option	Advantages	Disadvantages
CRL posted to CA web page	• Simple • Completely under CA control	• Users have to know web page • Applications have to be able to download and install CRL
CRL posted to X.500 directory via LDAP interface	• Standard interface, supported by all CAs and many applications • Centralized place to check revocation	• Directory must be set up and run by some organization (normally not CA)
Revocation information posted to OCSP server	• Somewhat more flexible than previous other approaches • Growing in popularity	• OCSP not yet fully supported by all CAs or applications (new standard) • Reliers must go to OCSP server each time a check is desired
Temporary hold (suspension) of certificate	• Allows quick but reversible action • Allows time to consider exactly what to do	• During hold, certificate is reported as revoked • Not all CAs may support

Other revocation requirements

The State may also wish to place some other common revocation requirements, such as those required by the Universal Postal Union (UPU), which include:

- Notifying the certificate owner (user) that his certificate has been revoked
- Ensuring that the RA verifies revocation requests (if the CA does the actual revocation)
- Archiving CRLs for at least 10 years
- CRL is published daily

The State may also wish to initiate helping a user to obtain a replacement certificate after the original certificate is revoked, if the user still has a valid need for a certificate. This could be done by sending a notice to the user.

CRL format

The list below shows the fields in the industry-standard X.509 version 2 CRL. There are basic fields and extensions for the entire CRL. There are also fields and extensions for each revoked certificate listed in the CRL. The following minimal set of fields is recommended. In particular, the CRL supports a reason code, which is a code taken from a small list of choices defined in X.509 and the PKIX (IETF) standards. CRLs do not support free-form reasons, although the CA database might. The UPU certificate policy requires the following fields or extensions (which are all non-critical):

- Version ("1" for X.509 version 2, because the count started at 0)
  - Signature algorithm ID
  - Issuer DN
  - This Update (Next Update is optional)
  - For each revoked certificate:
    - Certificate serial number
    - Certificate revocation date
    - Reason code
    - Authority Key ID

---

7 PKI-Enabled Applications

Overview

This section summarizes how the State of Iowa PKI might interact with various PKI-enabled applications of interest to the State. This includes how the State of Iowa CA can provide certificates to the applications, and how those applications use the certificates. The discussion is somewhat general at this point, summarizing the types of products available in the industry. Specific vendor products will be summarized and compared in a separate vendor product analysis, which is the next deliverable document specified for this consulting engagement.

Applications interaction with CA

One factor that determines whether certificates from a given CA will work in an application is the format of the certificates. This is usually easy to adjust using certificate template capability provided in the CA product. Another factor that can sometimes come into play is whether the CA must use a particular protocol to deliver the certificates. CA vendors can test various applications with their certificates, and if necessary modify the certificate format or protocol until they work.

The certificate format can be modified within broad limits to provide the certificate fields that are needed. Typically these are either part of the Subject Distinguished Name (the requester information) or one of a few standard extensions, such as the Subject Alternative Name. Typically, most applications can be supported by modifying the certificate templates, even if the applications have not been officially tested and qualified by the CA.

It should be noted that the Root CA certificate, and to a large extent also the subordinate CA certificates, do not require complex formats. They primarily include simple Subject Distinguished Name and Issuer Distinguished Name fields, and a few general policy-related extensions. Additional information, particularly application-specific fields, should be discouraged, because a root CA can not be modified and distributed easily just because a new PKI-enabled application is started.

The second factor -- certification protocols -- should not be significant for the State of Iowa CA, because standard clients such as browsers or e-mail clients are expected to be used.

Browsers and web servers (SSL)

Most users, at least the public, will be using web browsers as their clients. Web browsers have become the de-facto general-purpose client for most Internet activities, a point which State agency interviewees understood. Therefore the primary applications, such as back-end system access control discussed next, should assume that browsers provide the client software. Security for browsers and servers is provided by the Secure Sockets Layer (SSL) protocol, which supports both authentication (via an initial signature) and encryption using a single public key for each participant. Encryption uses the public key only long enough for the browser and server to exchange a one-time-only symmetric session key, which allows faster bulk encryption than a public key. This symmetric key is not based on public/private key technology; rather it uses an single old-fashioned secret key held by both parties.

The agencies thus will probably use web servers (hopefully sitting behind a properly-configured transparent firewall) as the initial interface to users. They can be relatively passive, for example just passing on contact to a back-end system. However, web servers can also be more active elements, using either built-in capability or additional plug-ins. Built-in capability includes certificate verification, for example (discussed further in a moment). Plug-ins add functionality, sometimes significantly changing the web server's behavior, while still holding to the standard browser/server front end interface model.

It is also possible that plug-ins can be used to augment the capabilities of standard browsers. Plug-ins can add functionality such as increased security, pre-programmed options, data forms specific to that system, etc. while still keeping a familiar web browser user interface. This may not be acceptable for systems to be accessed by the general public, who may be unwilling to install additional software. However, users who are government employees may have plug-ins installed by their IT staff. Even some public users may be willing to install plug-ins if it is easy to do and gives them access to a system perceived as high-value.

Access control - General

The most critical application identified during interviews with staff members of various State agencies is access control. This is required as a consequence of the recently-passed electronic commerce bill, which requires public access to State functions over the Internet using digital signatures. Most agencies expect to use a client/server architecture to provide these services to the public or to State employees, most likely with browsers as the clients and firewall-protected web servers as the servers.

The term "access control" actually encompasses several different security services, including authentication, data confidentiality, data integrity, management and enforcement of policy, and audit. These services may be delivered at many junctures between the desktop and the backend: by desktop security products, by firewalls, by virtual private networks (VPN), by web servers, by operating systems, or by applications. The range of security services required, the range of products to deliver those services, and the heterogeneity of enterprise environments all make access control a complex problem. There are many vendors and products, each covering different – though often overlapping – requirements.

Access control can be described as "coarse-grained" or "fine-grained", depending on the granularity of the resource being accessed. For example, firewalls and VPNs can implement coarse-grained access control by controlling which hosts a user can connect to, whereas a database application can implement fine-grained access control by controlling down to the level of fields in a database record. For some customers, access control requirements may be satisfied without the use of a special-purpose access-control product. The access control functionality in existing web servers, databases, or other applications may suffice, and the CA may need only to provide strong authentication via certificates.

The following sections summarize the leading access control options.

Access control: Standard web server and certificate verification

For strictly web-based access, one option is to use the access control functionality built natively into the web server. Some of the web servers that provide this capability are

- Native Netscape Enterprise Server
- Native Microsoft Internet Information Server (IIS)
- Apache HTTP Server

These are not usually sufficient for a complex network, because they protect one web server at a time. For centralized control and policy enforcement, scalability, and ease of use, one of the options in later paragraphs below is probably needed.

At a minimum, users must be screened to ensure that they belong to the system. This can be done by having the web server verify the certificate presented by the user's browser or other client. This includes several steps:

- Verifying first the user's signature on session data, for example using the standard SSL protocol.
- Verifying the online CA's signature on the user's certificate, using the CA's signature which is known to the server or is presented along with the user's certificate.
- Verifying the Root CA's signature on the subordinate online CA's signature, using the State of Iowa Root CA certificate which should be installed in the web server.

It should be noted that just verifying the user certificate up to a trusted root in this way (called "chaining") is not generally sufficient to ensure proper access rights. Other systems will have applications with certificates signed by the State root, of course. It is possible that an agency could have its own subordinate CA, and the web server can be set up to only trust that CA (not just the root). However, even here there is a risk that an intruder could somehow obtain a certificate from that CA.

Access control: Identity verification at back-end

It is best to identify the user from information, such as name and birth date or other ID, in the certificate itself. Once a user's identity is confirmed, his/her fine-grained access rights can be determined. It might be simplest to do this by reference to the back end database beyond the firewall and web server. This can work in a client/server environment in which the server includes access rights as part of its data. This has the advantage that access rights can be changed or cancelled by the agency's back-end system at any time, for example if the user's circumstances change. The functionality can be performed with a standard web server, with the back-end system doing the fine-grained access control. This does require some additional programming in the back-end system. It also means that users will get at least some logical access into the back-end system, even if they are not allowed to access sensitive data, which might conceivably be used in a denial-of-service attack.

Special access control products

Some access control products further allow the web server to identify users based on information in their certificates, so the web server itself can allow access only to appropriate selected files. The files shown to the user, and thus the web page data and hyperlinks the user can see and access, can be specific to that particular user or class of users. This is very flexible, and the products are designed so it is easy to set up fine-grained access to resources (typically files) based on the user's identity or a group to which he/she belongs. It would require buying one of these products.

Such products include SiteMinder by Netegrity and getAccess by Encommerce, among others. They allow system administrators to set up access control lists at the web server. These lists work much like access control list methods used in Unix or Windows NT systems, including grouping of users into similar classes to ease the administrator's task. Then the web server, in concert with one of these products, can decide which files can be accessed by the user. Inappropriate users will not even be passed to the back-end system.

Summary of access control options

Option	Advantages	Disadvantages
Standard browser / web server functions	• Simple; built into browser & server • No special software to install & learn	• Minimal security; basically only coarse-grained certificate verification • Extensions require plug-ins • Attackers with a plausible certificate can access back-end
Access decision at back-end	• More security and control, without requiring browser modification • Access rights can be changed at any time	• Requires additional development at back-end • Attackers with a plausible certificate can get slight access to back-end
Special access control products	• More security and control • Generally sit at web server • Flexible; access rights can be changed at web server via access control lists • May be no need to modify back-end	• Additional cost • Additional work to install plug-ins at web server and possibly web browser

Secure e-mail (S/MIME)

Virtually all modern e-mail systems now support certificate-based authentication and encryption via the S/MIME standard, which is also supported by CAs that provide S/MIME certificates. The e-mail system may be a stand-alone email product, or may be associated with a browser package (such as Netscape Communicator or the Microsoft Internet Explorer / Outlook combination). S/MIME certificates are much like browser certificates, typically just adding the user's e-mail address in the Subject Alternative Name extension. (Older versions put the user's e-mail address in the Subject Distinguished Name, and a CA may also put it there for backwards compatibility.) CA vendors routinely test their certificates with major e-mail systems and products, particularly those by Netscape, Microsoft, and Lotus (Notes). It should be noted that in the past, Lotus did not do a good job of bringing Notes into compliance with generally-accepted e-mail standards, and some lingering problems may still remain with the CA interface for getting a certificate or the certificate format.

Secure e-mail packages normally give the option of using a single key pair and its certificate for both authentication (via a signature) and encryption, or separate key pairs for authentication and for encryption. (As with browsers, encryption uses the

public key only long enough for the parties to exchange symmetric keys.) It is simpler to use a single key pair for both signing and encrypting, because users only have to keep track of one certificate. A single key pair per user also means that once one user sends a signed message to a second user, the second user has the first user's public key (from the certificate, used for verification), and thus can use that same public key to encrypt messages back to the first user. A separate encryption key pair means that a directory must normally be used to obtain other users' public encryption keys. One concern with using a single key pair for both signing and encrypting is a highly theoretical attack that takes a known plain text and enormous memory, and the fact that usage of signature keys and encryption keys are somewhat different and might be better kept separate. Neither of these concerns are taken very seriously by cryptographers. Another more serious concern is that if a system requires both legally-binding signatures that must hold up in court, and persistent storage of encrypted information that must be recoverable via a backed-up key, then a single shared key may be infeasible. This is because backing up a key for encryption reasons can invalidate that key for signature reasons.

Software packages are available to add flexibility and power to the standard e-mail packages above. They may support more functionality such as a better user interface, or allow more e-mail packages to be supported. Some examples are Baltimore Technologies' MailSecure, Entegrity's AssureMail, or Entrust/Express. As one would expect, these products integrate well with the vendors' other PKI offerings, but generally also work with other major CA's certificates.

Summary of secure e-mail options

Option	Advantages	Disadvantages
Standard e-mail products	• Simple; basic PKI built into e-mail products already • No special software to install & learn	• User interface sometimes weak • Some
Special e-mail enhancement products	• More functionality and better user interface • Easier to interface with some e-mail packages and/or CAs	• Additional cost • Additional work to install software
Single key pair (and certificate) for signing and encryption	• Simpler to keep track of keys and certificates • Allows message recipient to encrypt messages back to sender thereafter	• Some theoretical issues with possible future attacks with known message text and enormous memory
Separate signing and encryption key pairs (and certificates)	• Possibly slightly more secure against future attacks • Separates signing and encryption functions, with their certificates	• Slightly more complex for users: requires them to access a directory to get others' encryption keys

VPN (IPSec)

The State of Iowa may wish to provide virtual private network (VPN) certificates from the State of Iowa CA. Virtually all VPN products now support the Internet Protocol Security (IPSec) standard, which includes certificates used by the VPN remote clients or hosts for setting up authenticated and encrypted Internet links. Note that a VPN host, which controls access to a network at the server end, may sit on a firewall, a router, or stand-alone hardware.

Both the certificate format and the method of getting a certificate into a remote client or host vary somewhat with different VPN vendors. A common certificate format is similar to a browser certificate but with the IP address in the Subject Alternative Name extension. (If dynamic IP address assignment is used, however, there is no fixed user IP address so an IPSec client certificate must contain other identifying information such as an approved user name.) The means of getting the certificate into the client or host may be to act like a browser as far as the CA is concerned, using one of the PKIX certification protocols, or to load the certificate as a PKCS#12 file obtained elsewhere. Unfortunately, Cisco has taken a somewhat non-standard approach to certification, with their own protocol (Simple Certification Enrollment Protocol or SCEP) and certificate format. However, given Cisco's dominance in the networking and Internet fields, most CAs now support Cisco's approach as well as the more standards-based approaches of other VPN vendors. The different vendor options will be summarized in the upcoming vendor survey document.

Form signing, encryption, document transfer

Stand-alone documents or web forms can be signed and encrypted to provide assurance of integrity, non-repudiation, and confidentiality. Until recently, most such products did not use public key methods and were not certificate-based. However, this is changing, and it is now possible to protect and exchange documents with the advantages of certified public keys. Document portection products include Baltimore's FileSecure, Aliroo's PrivaSeal, and AT&T's Secret Agent. Web form protection products include Baltimore's FormSecure and products from UWI.com. These products support both authentication (via signature) and encryption, supported by signature and/or encryption certificates, using keys which may be the same or different.

Directories

X.500 based repositories, called directories, are used to store and "publish" certificates and/or CRLs. Users or relying parties (who need to verify certificates) can then contact the directory for a user's certificate or the latest CRL. The State of Iowa may want to use a directory; some State agencies probably already do. This will be separate from the CA, although CA vendors should be willing to recommend directories they know work well with their CA. CA products all use the standard Lightweight Data Access Protocol (LDAP) to interface to directories. The interface is usually fairly straightforward, but occasionally there can be a mismatch in the directory schema (essentially a subset of the Subject Distinguished Name) that requires either the CA or the directory to modify its interface slightly. Some examples of directories that have worked well with CAs include Netscape's Directory Server and the ISOCOR directory.

Smart cards and tokens

Smart cards can be used in a PKI to store private keys (and the corresponding certificates) in a more secure manner than the normal storage method: encrypted on a PC's hard disk. Ideally, the key pair should be generated on the smart card, and the private key should never leave the card. In this case, once a smart card is removed from a PC, the private key does not exist at that PC. This allows secure operation with multiple users on one PC; each user just plugs in his/her smart card when using PKI, then removes it. Smart cards also allow one user to have multiple PCs, each of which can have the user's private key only when he/she plugs in the smart card. Another recommended use for smart cards is to increase the security of Registration Authorities at their browsers, given the importance of protecting the certificate authorization process.

Most smart card vendors now support key/certificate storage with browsers. This is handled using software drivers and smart card readers that are provided by the smart card systems vendor. The CA does not have to be involved in this process; in fact the CA does not even know or care where the keys and certificates are stored. The only real issue is that the smart card must have enough memory to hold any necessary keys and certificates. (A typical key pair plus certificates takes about 2 to 4 Kbytes, and modern smart cards have 8 or 16 Kbytes of memory.) The smart card vendors who offer such products, and have proven interoperability with major CAs, include Schlumberger, Gemplus, DataKey, Litronic, ActivCard, and DataCard.

Other applications

Other applications were mentioned for possible future interest by some interviewees, but there is no immediate need and the State may or may not wish to include requirements to support them at this time. These include wireless applications, business-to-business document transfer, and more secure credit card Internet transactions. They were discussed, along with the other applications above, in the PKI Requirements Analysis documents, and are summarized briefly here.

Wireless (WAP): Wireless applications will allow a mobile phone with a small graphics screen to access the Internet almost like a miniature browser. The set of protocols that supports this, collectively referred to as WAP (Wireless Access Protocols), are now in development and products are being prototyped. The protocols parallel the familiar PC-oriented Internet protocols like TCP, IP, HTTP, etc., but are optimized for wireless products with their more limited speed, memory and display capabilities. Baltimore Technologies is particularly active in the WAP field, both setting standards and developing supporting products.

Business-to-business document transfer (EDI, XML): The most general method for passing business-critical documentation and data uses Electronic Data Interchange (EDI) and/or EXtensible Markup Language (XML). XML is somewhat like a more general and extendable equivalent to HTML, which is used by browsers, but allows industry-specific extensions. EDI and PKI standards and products are slowly converging on joint standards, via techniques such as the EDIFACT studies, and the X.509 standard now supports placing an EDI name (ediPartyName) as a field in the Subject Alternative Name extension. Baltimore Technologies now also offers a PKI-enabled XML product called X/Secure.

Secure credit card transactions (SET): MasterCard and Visa jointly developed the Secure Electronic Transaction (SET) protocol to protect credit card purchases over

the Internet. It allows much greater protection against attack, and more privacy for users, than the present alternative of using a "shopping cart" program at the web server protected only via standard SSL. SET is being used extensively in Asia, and increasingly in Europe. It is much less common in the U.S., primarily because U.S. credit card liability limits are set at only $50 by law, so users and merchants are not concerned enough about credit card fraud to install new software..

8 Other CA Functions

Overview

This section covers miscellaneous architecture elements that do not fit with the previous sections' topics.

Backup

The CA system should be backed up periodically. A good practice is to back up the CA's database every 8 hours, and save the backups on tapes or other media that are removed daily and stored outside the CA facility.

The CA keys, including the State of Iowa root, should be backed up in case of a failure of the keys or cryptographic unit that contains them. Ideally, the backed up key should be split into multiple parts which can be distributed to different officials.

Network management tools

We do not recommend the use of OpenView or similar network management software for backup or "write" functions due to serious concerns about its security in a CA high-security environment. If such software is used, it should be carefully designed to avoid using the "rlogin" installation option (or any other unnecessary communication protocol), functionality should be read-only, there should be no connections to outside networks (e.g., the Internet), and the agents should be as simple as possible so their functionality can be verified to be safe. The primary intent of these recommendations is to avoid the possibility of a "back door" entry into the heart of the CA.

Monitoring

The CA should frequently run network monitoring software to help establish CA security. Such software should both probe the CA's protections to determine if they are working properly, and monitor network traffic to detect suspicious activity that may show the CA system is being attacked or probed.

CA certificate distribution

The CA certificates, at least for the State of Iowa Root's public key, must be exported so that applications can verify the CA signature on certificates. The CA must have procedures for doing this.

Certificate storage and dissemination

Typical CAs store certificates (along with CRLs and audit logs) in an internal database, such as Oracle. This database is used to provide certificates and CRLs to CA web pages for download, to provide standard reports to CA operators, and perhaps to support custom SQL queries about certificates or audit data. The CA should also be able to export certificates and CRLs to an LDAP-compatible directory. Organizations are then able to collect and store certificates or CRLs as appropriate.

9 Multi-Certificate System Design

Overview

This section discusses issues relating to design and configuration of the CA and certificates to support the State of Iowa PKI systems. It addresses several requirements from the PKI Requirements document.

Certificate format

The X.509 standard and major CAs support all certificates that the State of Iowa is likely to need. The PKIX (PKI extensions) standard includes the full X.509 standard and adds some new extensions. Most major CAs support all X.509 and most PKIX fields. It might be simplest to just ask prospective CAs which PKIX extensions are not supported fully, and see if any important extensions discussed below are not supported.

The simplest certificates are those used for browsers (SSL) or email (S/MIME) clients, which may have no extensions and put all information in the Subject and Issuer distinguished name fields. S/MIME certificates are much like SSL certificates, with the inclusion of an email address in the Subject Alternative Name set of extensions. More complex certificate formats may require some extensions, as described in paragraphs below.

A typical CA can support one or more certificate types and formats, including all those listed in the PKI Requirements document. It can do this by allowing the CA's home page to have multiple hyperlinks. Each hyperlink sends the user to the proper certificate request page, with the correct web form and certificate format for that type of X.509 certificate. Certificate data is specified by tailoring the web pages. Certificate formats are specified by certificate templates that can be edited for a given system. One subordinate CA may support one or multiple certificate types, depending on the architecture built into the CA vendors software.

Some certificates may certify keys intended only for signature or for encryption. These may be defined by the certificate template, using the key usage extension discussed in the next paragraph, and would require separate hyperlinks to different request pages. Most CAs now support dual certificates, in which a single certificate request can be used to obtain matched signature and encryption certificates with the same user information.

Certificate extensions - general

As indicated above, some certificates will require standard X.509 extensions to support special features. These extensions are defined either in the X.509 standard, or more recently in the PKIX standards of the Internet Engineering Task Force (IETF). Extensions are normally defined by a template for that certificate type. The CA may include a template editor that is used to format these templates. Extension contents may be set by the template if they have fixed values, or may require input from the user via the certificate request form. Browsers should be able to tolerate extra extensions, and will just ignore any extensions they don't understand, as long as the extensions are not marked as "critical" in the certificate encoding. But some clients could process the same certificates, including the extensions, to perform additional functions. (Applications that do not understand an extension marked as "critical" in the certificate are supposed to reject the certificate, so critical extensions are seldom used except in systems such as SET where the same organization controls both the CAs and the certificate-using applications.) Extensions that the State of Iowa may want to support are listed below. It is likely that not all will be needed, but it is best to require a CA to be able to support as many as possible, just in case a need arises.

Certificate extensions - details

All original X.509 extensions, and the common PKIX extensions, should be supported. At least the following extensions should be supported if possible. Those that are most likely to be needed are so noted in the description.

- Key Usage: Specifies whether the certificate is used for signing and/or encryption, among other uses. (Can also include non-repudiation, encryption versus decryption, and key agreement.) Typically set by template This is an important extension if separate signing and encryption certificates will ever be needed, as the State is likely to require.
- Basic Constraints: Defines whether the certificate is for a CA or an end entity (user), and includes maximum verification path length (number of CA hierarchy levels). Typically set by template. Required for CA certificates but optional for end-entity user certificates.
- Key IDs: Used for certificate path chaining. Desirable but probably not essential. Two types:
  - Subject Key ID: A hash (compression) of key material, used for root or subordinate CA certificates. Computed based on certificate data.
  - Authorizer Key ID: A hash, or an issuer name + serial number. Inherited from the parent CA.
- Subject Alternative Name: Used for any of about 9 special user-specific "names" such as email address, IP address, URL (web address), EDI name, directory name, etc. Set by the user in the certificate request. Now required for e-mail and VPN certificates, to hold the e-mail address or IP address respectively. (Earlier applications just put these addresses in the Subject Distinguished Name, but this is no longer acceptable for new clients.)
- Issuer Alternative Name: Used for special "names" for Issuer, similar to Subject Alternative Name above. Typically set by CA template. Not used as often as Subject Alternative Name.
- Netscape Certificate Type: For Netscape browsers or servers. Typically set by template. Not required, at least according to Netscape, but may be a good idea to require just in case future browsers expect it.

- CRL Distribution Point: Tells applications verifying the certificate where the latest CRL can be found. Typically set by template. Many applications can now use this if provided; should be supported.
- Various legal or policy-related extensions, listed below. Typically set by template.
  - Policy Identifier: An Object Identifier (OID) code for a policy that should be met. Includes two sub-fields below. It is recommended that the OID and both sub-fields be supported.
    - UserNotice: A short statement (200 bytes) on how the certificate is to be used.
    - CPSUri: A URL pointer to a web page with the full legal or policy statement
  - Netscape Comment: A short statement, only for Netscape certificates. Recommended, partly because warning can be placed here that are viewed when the browser "view certificate" button is clicked.
  - Policy Constraints: Forces all subordinate CAs to include a reference to an acceptable policy identifier. Probably not essential.
  - Name Constraints (CA certificates only): Forces restrictions on legal names of CAs in the verification path. Probably not essential.
  - Terse Policy Statement: A short statement, only in SET certificates. Not needed unless the State becomes interested in SET, in which case it can be subsumed into a general requirement to be "SET compliant".

10 Other Design Issues

Overview

This section discusses miscellaneous issues relating to design and configuration of the CA and certificates to support State of Iowa systems. It addresses several miscellaneous requirements from the PKI Requirements document.

Pilot systems

It is a good idea to start with a small and well-controlled pilot PKI system before launching a large critical PKI application. A good pilot may include between 100 to 1000 users. This allows the State to test PKI applications and the CA in a less stressed environment. In addition, the State can become practiced in the operation of a CA facility, and can initiate and write up realistic operational procedures. In the long run, a good pilot generally saves time compared to jumping directly into a full operational environment. (Going directly to full operation has been done successfully, but it required a very skilled systems engineering vendor with overall responsibility.)

Root keys and certificates

As discussed in the PKI Requirements document, there are some options for providing a root to applications. These are listed below.

- The State of Iowa creates its own root and provides it to the public: The root, or for that matter any subordinate Iowa CA certificate, can be provided on a web page, for users to embed it in their browsers themselves. (It is set up with a MIME type that tells the browser to install it.) The user is asked whether to trust the root or CA certificate as part of the installation process. This is a fairly easy process, but can be intimidating because Netscape and Microsoft browsers show somewhat confusing windows advising about security implications.
- The State of Iowa creates its own root as above, then has it embedded in major applications such as browsers and servers. This is not easy; it is difficult for CA operators to complete the necessary negotiations with Netscape and Microsoft in time for each new release. Even some CA vendors have not completed this process. It is not clear if a U.S. state will have an easier time of this.
- Tying the State of Iowa root to an existing root which is already embedded in major applications: The Baltimore Technologies (formerly CyberTrust) OmniRoot can be used for just such an application. The drawback is that it can appear that the State of Iowa has lost some control to a higher commercial entity, although in fact the subordination is only a technical matter. The embedded root vendor should put only reasonable restrictions on the State, which it should already be meeting for its own reasons.

Summary of root CA options

Option	Advantages	Disadvantages
State root, provided for public to embed in applications	• State retains full control of its root	• Users have to embed root in applications (e.g., browser); can be confusing
State root, embedded in applications by State	• State retains control of root (just has to negotiate where it is embedded) • Simpler for users	• State must negotiate with application vendors (e.g., Microsoft and Netscape); can be difficult, time consuming and expensive
Tying state "root" to commercial root already embedded in applications	• Simpler for users • Less trouble for State than embedding their own root	• State has to coordinate with both application vendors and CA root vendor

Cross-certification

Most CA product vendors now support cross-certification with each other, at least as far as technical matters are concerned. It consists of each CA submitting an industry-standard PKCS#10 certificate request for its root to the other and being given a signed X.509 certificate in a PKCS#7 "wrapper". Then users in the other CA's system will be able to verify the first CA's certificates. Business issues have generally been trickier in commercial applications where this would require competitors to certify each other as meeting acceptable standards. However, state and federal governments are accustomed to cooperating and treating each other as equals. Therefore such a cross-certification capability should be required, along with some evidence of cross-certifications performed in the past. Note that the signed CA certificate is no longer a true root, which is defined as a self-signed certificate (the Issuer and the Subject having exactly the same Distinguished Name structure). Either the original self-signed or the externally-signed CA certificate can be used as appropriate; they are different certificates but use the same key pair.

Directory & LDAP

CAs can generally support a Lightweight Directory Access Protocol (LDAP) interface to an X.500 directory, for the purpose of supplying certificates (and CRLs) to the directory. However, there are some formatting glitches that can occur, so it is not always trivial to actually get the LDAP interface to work with a specific CA and directory. Prospective CAs should be asked to describe the types of directories with which they can interface via LDAP.

Tokens

The State of Iowa may wish to support user certificates on hardware tokens such as smart cards as an option. It is also recommended that remote RAs connecting with the CA over the Internet be required to have special RA certificates that are stored on tokens. This can generally be done with existing browser capabilities, using token vendors' hardware and software, and should be transparent to the CA.

Several vendors offer smart card interface kits for Netscape or Microsoft browsers. They consist of installation software, smart cards, and a smart card reader, and presently cost about $100 US. (This cost should drop as more kits are sold.) The browser and smart card interface do all the additional work, and the CA does not have to change anything in its end of the certification process. Once installed, the browser then has the option of using the smart card, rather than the usual hard disk

interface, to store keys and certificates. The smart card interface shows up in the browser user interface as an alternative cryptographic module, and the user is asked which module should be used whenever a certificate must be chosen. When the smart card is removed, neither the private key nor certificate exist on that platform. Baltimore Technologies (including the former GTE CyberTrust) has tested several vendors' products and shown interoperability with several, including Gemplus, DeLaRue, Sclumberger, DataKey, and others.

A lower-cost token alternative to smart card that is beginning to appear is the use of "dongles" that plug into the Universal Serial Bus (USB) connector that is now standard in almost all new PCs. These dongles cost about the same as smart cards, and operate in much the same way. However, they avoid the need for a card reader, thus saving about $50 per user.

Redundant system

If the State of Iowa runs its own CA, it may wish to use redundant equipment, where most CA and network components can be duplicated and used as spares, or as co-processors to improve performance. This will double the hardware cost, at least. If the State outsources the CA functions, the CA service vendor should be required to have redundant components for all major functions.

Help desk

A help desks is essential for any large system, to support users who are having difficulties. There are two types of help desk functions, which can be handled separately.

- Problems with the web interface (e.g., browser) or with getting a certificate. These tend to be similar across multiple systems, and the State may wish to centralize this type of support. The commonest problems in our experience are forgetting passwords, and being confused by the many browser windows that pop up when getting or using certificates.
- Problems with the specific agency application that is using certificates. These tend to be different for different agencies, or even different applications within one agency. Examples might be problems getting documentation, or unusual circumstances not foreseen when the system was set up. In most cases, the agency or agencies running a PKI-enabled system would have to help users handle these problems.

Other requirements

There are miscellaneous requirements that the State of Iowa should meet, and require CA vendors to meet. They include:

- Various certificate format and revocation requirements, already covered in the corresponding sections above
- Ensure uniqueness of Subject Distinguished Names within a CA Issuer
- Specify maximum certificate validity periods
- Extensive audit events
- Digitally signed audit logs (typically daily)
- Backup archives of audit, CRL, and certificate data
- Providing CA root key (for example for download into a browser or as a file)
- At least 1024-bit RSA keys for users and subordinate CAs and 2048-bit RSA keys for the Root CA

- Two (or more) person control of CA's private keys
- Secure backup of CA keys
- CA quality control system (e.g., ISO 9001 and external audits of CA operations)
- Secure firewall and security intrusion checking

11 CA Key Management

Overview

This section provides a sample key management plan for handling CA keys and certificates, and particularly a CA's all-important private key. It summarizes in one place how the certificates used by a CA are created, and how to handle the keys the CA uses (for example for certificate signing). Prospective CA vendors should be required to support operations like those described here, in order to ensure that the CA operator (whether the State of Iowa or a CA service provider) can securely manage CA keys.

Scope

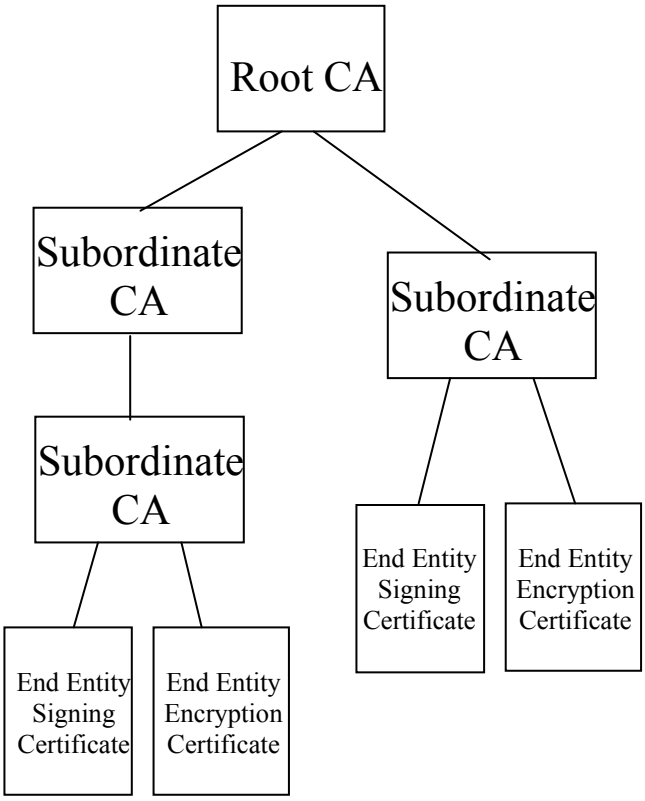
This plan describes cryptographic key and certificate handling for the State of Iowa CA. It focuses primarily on keys and certificates used within the CA, rather than end entity keys such as for users. It also summarizes the extensive security measures to ensure that key compromise or mishandling is extremely unlikely.
It covers the following topics:

- Key and certificate issuance, including root hierarchy
- Key and certificate renewal
- Backup and splitting keys
- Disaster recovery
- Key compromise, revocation and replacement of keys and certificates
- Security protections

State of Iowa CA Keys and Certificates

CA hierarchy

CAs that are used with large or high-security systems normally support the creation of hierarchies of CAs. There is a root CA that is kept offline, whose primary function is to certify subordinate CAs. The root CA is self-signed, which means that since there is no higher-level CA, the root's certificate is signed by the private key of the same key pair that is being certified. (The subject and issuer names are also identical.) The subordinate CAs are what actually provide certificates for end users (end entities). End entity certificates might be used for signing, encryption, both, or other functions. This architecture has at least two advantages: (1) It allows for distributed certificate management which shares the certification load across multiple CAs that all still tie to a common root; (2) It protects the root from direct attack over the Internet. The figure below shows a sample CA hierarchy, consisting of a top level Root CA with a self-signed certificate, subordinate CAs certified by the Root CA, and End Entity certificates issued by the subordinate CAs. As it shows, the number of hierarchy levels may vary, but there is usually at least one subordinate CA so that the Root CA does not have to be on-line and signing end entity certificates.



Sample Certificate Hierarchy
(please view in Page Layout mode)

Root and subordinate keys

The Root key pair is the most important item in a CA and is necessary in order to create a hierarchy of trust. The State of Iowa CA must be capable of generating and supporting multiple root CA private/public key pairs and certificates (i.e., multiple CA hierarchies). For the purpose of implementing a CA hierarchy, subordinate CA private/public key pairs and certificates are generated under a designated root key. Multiple subordinate CAs can be supported under a single root key.

In the CA hierarchy model, a subordinate CA provides certificates to end entities, in response to a certificate request containing the user public key. In most cases, the user retains its private key. One exception may be issuance and distribution of keys to end entities on floppy disks or smart cards, if that is to be supported.

The State of Iowa must decide what its root key will be. This topic is discussed in the PKI Requirements and other sections of this architecture document. There are basically two choices:

- Self signed State of Iowa Root Key
- Chain up to an existing known and trusted Root (e.g. Omni Root)

In the first choice the State will be responsible for embedding the root into end entity software (i.e. browser and servers) or provide some means for the user to download

the root. This process is not trivial and does require the end user to perform some additional tasks. The second choice is desirable from an operations standpoint because end entities will not have to perform any extra steps. The roots are already embedded in the software and certificates can be used immediately after issuance.

CA signing keys and certificates

CAs at all levels have their own key pair for signing and verification. Each CA uses its private key to sign the certificates it creates. The matching public key for verification is provided to recipients in a CA certificate. By definition, root CAs sign their own certificates, but all other CAs' certificates are signed by the CA immediately above them in the hierarchy.

User key types

The State will provide its CA public key certificates to entities such as users (or possibly other CAs) when requested. It also obtains user public keys for required data encryption or verification operations. Typically, users are responsible for their own private keys and they are not a CA concern.

However, in some cases, the users may request the State to generate and issue key pairs as part of an RA-assisted certification offering. In this scenario, the user keys will be generated in the secure facility and kept protected both physically, in safes, and electronically, with passwords or temporary certificates, until delivered to end users.

State of Iowa Key and Certificate Issuance Hierarchy

Key issuance

This section steps through the normal initial setup of keys and certificates for the State, ending with a fully functional system. Although each entity creates its own keys, it depends on a CA above it to sign its certificate, so certificate issuance proceeds from the top down.

Root setup

The State of Iowa Root CA is at the top of a hierarchy. Access to it is very tightly controlled. Using two-person control (i.e., with an observer) and access control methods that vary with different CA vendors, the operator will activate the off-line Root CA computer platform, initialize the root crypto module, and create the root signature key pair and self-signed certificate on the root crypto module. The private key stays in the crypto module, as always, and is used to sign lower-level CA certificates. The root certificate is also transferred to the on-line CA computer platform, for example via a floppy. The root public key can be distributed to users in software, for their verification of certificates referenced back to the root. The root public key may also be provided over the Internet, in a Web page or embedded in standard browsers.

CA Key and Certificate Renewal

Renewal summary	Normal key and certificate renewal for CAs is like the initial certificate issuance in most respects. CA keys and certificates are renewed before their certificate expires, by bringing the appropriate crypto module to its superior (off-line) CA for updating. The Offline CA platform commands the CA module being renewed to create a new key pair, and makes a new certificate much like the old one except for the public key and dates. The CA signs the new certificate, which is stored on the workstation and renewed module.
Crypto-periods	The length of time during which keys are valid is flexible. This can be operationally set as required. The root keys will be valid for several years (e.g., 5-15 years). Other CA signing keys can be changed as determined best, since all that is necessary is to start signing certificates with the new key in the certification path to root. Changeover intervals may be about 10 years for the Root CA, and about 5 years for subordinate CAs, assuming user certificates are marked as valid for about 2 years.
Root key distribution	Once the new root certificate is distributed, it can be distributed to any applications such as browsers that may need to use the root key for verification purposes. All certificates may be accompanied by an internal certificate chain back to the root, although users may cache known good public keys to avoid verifying all the way to root. The new certificates and their associated keys can be used immediately by any user who has received the new root certificate. However, old root certificates for verification must be held and provided to users indefinitely, in case any signatures signed with the key might have to be verified later.
Root renewal	Renewal of a root key pair and distribution of its public key certificate is a critical function. All other certificates are signed by reference to their root certificate, directly or indirectly, so a change to a root affects all other signing entities below it. Whenever a new root key pair and certificate are created, the new public key (in the CA certificate) must replace the old public key in all relevant applications. A CA certificate must be valid, for verification purposes, for some time after the CA stops signing with that private key. Therefore, the CA must have a shorter *key* validity period than the *certificate* validity period. In particular, the CA should refuse to sign any certificates that have a validity period longer than the CA key's validity period.
Renewal location (off-line CA)	CAs are updated at the Offline CA, which means that the CA crypto module must be briefly removed from the on-line system. Incoming certificate requests for that service must be routed to an alternate CA module or queued.
User renewal	Renewal of user (e.g., user) certificates is outside the scope of this section. The PKI architecture document describes the renewal of end entity user certificates.

Backup and Recovery Due to Failure

Problem	The State of Iowa CA must be prepared to recover from loss of cryptographic capability, including destruction of crypto modules and keys. This could be the result of a disaster that damages hardware, or a failure of one or more crypto modules because of a local problem. It does not include situations where keys may have been exposed and compromised; that is handled in following sections.
Crypto module key backup and recovery	The crypto modules that contain CA keys must be able to back up those keys onto another crypto module, in case the first module fails. This is particularly important for Root CA crypto modules, because unplanned replacement of a root key is almost impossible. This is largely because the root certificate has been distributed to many applications and users, who are not expecting to have to replace it before its normal expiration date. This backup process must be secure. At a minimum, the private key must never be exposed when it is being copied from one crypto module to another. This is done by encrypting the private key in a special "key encryption" key known only to the recipient crypto module. The backup module can then be put in a safe place, preferably a safe that requires multiple independent combinations to open. It is also desirable to be able to split the root key into multiple parts, distributed to separate individuals so that no one individual can recreate the private key inappropriately. In this case, if the original key is ever lost, each key holder (or perhaps a specified subset of them) must bring their portion together to recreate the original key in a new crypto module.
Off-line CA key replacement	If for some reason the backup module is not available, replacement with a new key is necessary. The most difficult key to replace is a root key, because all certificates refer to it. Thus all users must be provided with the new root key's certificate and hash before they can verify any new signatures. If any other CAs are destroyed and no backup is available, a replacement CA would just be created with a new key, which can be used immediately after a signed certificate is created.
On-line CA recovery	If an on-line service CA is destroyed, it can be replaced and used immediately by just creating new crypto modules and keys (as long as no compromise occurred). The standard module and key creation process defined in a CA operations manual (or analogous documentation) is used. Alternatively, a backup of the on-line CA can be prepared in advance; this means the signing key will not change. Note that even if a key is destroyed, old certificates that used it in the signature chain are still valid, and can be verified using the certificate chain up to the root.
On-line redundancy	To minimize the impact of damage to on-line CA, multiple on-line workstations and crypto modules can be used. Thus loss of any one workstation or module will not shut down the entire State of Iowa PKI system, but rather will just have the effect of temporarily reducing capacity until the failed unit can be replaced.

Catastrophic Failure

The State of Iowa may wish to have a full backup CA facility in case of a failure of major portions of the original CA, for example due to a natural disaster. This might be placed at the STARC Armory, as discussed in the PKI Requirements document. The essential CA components could be recreated there from backups, using tapes of software and database contents, and duplicate cryptographic modules.

Key Compromise, Certificate Revocation, and Replacement

Problem

The following paragraphs deal with recovery from a possible CA key compromise, or other situations where a CA certificate must be revoked. They cover replacement of internal keys used by the State of Iowa, security issues, and the impact on users.

Impact

Unlike disaster recovery, the CA key would now be suspect, so it would have to be revoked and a replacement generated. This would be a very serious problem, to be avoided at all costs, as discussed in the following paragraph. Even aside from the public relations impact, it could be extremely difficult to replace all the suspect subordinate end-user keys that are dependent on the CA key for their trust. For CA signature keys, all impacted users would have to be notified of the compromise and the time a key became suspect. Impacted users include all users of certificates below the exposed key in the certificate-signing hierarchy tree, whose certificates were valid at the time of compromise. They would have to obtain new certificates for any keys they need. In a worst-case scenario, this could mean revoking all active user certificates and then renewing them via new certificate applications, an immense undertaking.

Protection against compromise

Since recovery from possible compromise of a major CA signing key is very disruptive as discussed above, extreme precautions are taken to ensure that such compromise is highly unlikely. Because there should no way to read a key out of a crypto module, other than in encrypted form for key backup into another module, physical control of crypto modules at all times can eliminate the possibility of compromise. Security procedures are a topic in themselves, typically covered at length in a separate security procedures document. In summary, they should include: multi-person control using multi-lock safes; two-person operational procedures during module exposure; locked rooms and cabinets for on-line crypto modules; tamper-evident crypto modules; and restrictions on backup. High-level keys such as root keys are handled only at an off-line workstation, and such handling is a relatively rare event (for example creation of a new on-line crypto module). This is carefully observed and controlled.

Compromise evaluation

In the unlikely event that a key is exposed to possible compromise, the first response would be to evaluate the potential damage caused by the exposure. Audit records, operator and guard logs, and any other relevant records would be examined. It might be possible to show conclusively that although correct procedures were violated, the redundant security protections still averted compromise. In this case, no response would be needed, other than correcting the situation to ensure it cannot happen again.

New key creation and notification of users	If the evaluation concludes it is necessary, a new key would be created, and the next lower layer of CAs would have to have their certificates re-signed by this new key. Also, impacted users would have to be identified and notified. The certificate database might be searched to determine this information. For a root key, notification would be broadcast by any appropriate means, which could include the Internet (Web or email messages), contacting application vendors, news media, or any other mechanism including those used to distribute the key in the first place. The notification would include the effective time of the compromise. All impacted users would have to stop using keys whose certificates were signed by the key after compromise, and obtain new certificates signed with the new key.

Security Protection

Overview	Careful handling of secret keys and public key certificates is vital. Compromise of a CA private key (via loss of control of its crypto module) would put all lower subordinate keys at risk, and could require extensive recovery measures. Improper handling of a public key certificate is less damaging to overall system security, but would still be disruptive to the user and damaging to the CA's reputation. This section summarizes protective measures that ensure such security failures are virtually impossible. They include crypto module security, facility physical security, personnel access, software, and operational policies.
Cryptographic module security	The CA cryptographic hardware modules protect the critical private keys. The module design keeps the keys inside at all times, internally performing all private key operations: generation, signing, decrypting, or encrypted backup. A PIN, unique to each module, must be entered to activate the module. The modules should be tamper-evident at least, with a sealed case. Some degree of tamper resistance is also desirable. Keys in crypto modules should be impossible to extract by just reading memory.
Physical security for crypto modules	Crypto modules should be locked in safes when not in use, or should have built-in protection equivalent to a safe. The upper-level CA-signing modules are only used a few times a year. They are brought out briefly to activate or renew lower-level CA modules at an Offline CA workstation, which is not connected to any network. Floppy disks are manually carried between workstations (“sneakernet”) when transfer of material is needed. Only lower-level Online CA modules are used on a continuing basis, but they should be kept in a dual-locked and alarmed room that protects the modules during operation. The doors and safes should require two locks, with keys or combinations held by different individuals.
Facility physical security	The entire CA facility must be located in a physically protected room, with a logbook, dual door locks, security camera, motion detectors, and other physical protection.

Operational procedures	Strict operational procedures are used. Two-person access is required, so no one person could endanger the system. Separate roles, such as security officer and CA operator, are enforced operationally and via possession of crypto modules, PINs, and passwords. All activity is audited and logged and frequently reviewed, to ensure that intrusion attempts, improper access procedures, or any other attempts to injure the system are detected and blocked. Personnel with clearances or background checks may be used for critical operations.
Backup	Protection against failure or physical disasters is provided by the use of redundant and backup capabilities, including the use of crypto module backup capabilities discussed earlier. Both the backup module and any split key devices are stored in separate safes in a different location. Backup is a relatively infrequent and very carefully controlled process.
Revocation and replacement	In the unlikely event a possible compromise occurs, despite the protection described above, the procedures to be followed are summarized in the Key Compromise, Certificate Revocation, and Replacement section above.
Cryptographic data protection	Communications security is provided by the use of signing and encryption as an integral part of the message protocols. Keys are periodically changed, with crypto periods that trade off exposure, potential damage of compromises, and operational impact of renewal. Common equipment (communications and security front end equipment) such as a firewall and web server ensures isolation of separate users by file access privilege mechanisms, and precludes attacks or interference coming from the Internet.